Ugo

1. Alice wants to send Bob a long message, and she doesn't want Eve to be able to read it. Assume for this scenario that AITM is impossible.
    a. Using Diffie-Hellman is a very simple and viable solution for this scenario. Alice doesn't want eve to be able to see what she's sending, and she literally can't if she is just a spectator. The only way for eve to be able to do anything here is if she steps into the exchange and starts tampering with data, which we've ruled out in this scenario.
2. Alice wants to send Bob a long message. She doesn't want Mal to be able to modify the message without Bob detecting the change.
    a. In this scenario, I believe public secret key encryption gets the job done here. One problem is that public secret key encryption is usually used for shorter messages, so what we'll do is shrink the message by using sha-256 on it and let that be our message M. The thing here is that she doesn't want the message to be tampered with and Bob not notice. Well, Bob will notice immediately because not only is he the only one who can decrypt Alice's message, but Eve couldn't do so even if she was in the middle. Alice will encrypt her message twice, once with here secret key and once with Bob's public key, M" = E(P_B, E(S_A, M)). Bob will be able to extract M from M"" by hitting it with his own private key, S_B, and then Alice's public key, P_A. Because we can assume everyone has kept their key's private and have the right public key, If Bob can't decrypt it, then he's attempted to interact with a faulty secret key from Alice. He'll then be able to run the same hash function on it and see the message.
3. Alice wants to send Bob a long message (in this case, it's a signed contract between AliceCom and BobCom), she doesn't want Eve to be able to read it, and she wants Bob to have confidence that it was Alice who sent the message. Assume for this scenario that AITM is impossible.
    a. In this scenario, with AITM impossible, we can combine the use of public-secret key pairs along with the sha-256 cryptographic hash function. Alice can create a digest of the message using sha-256, D = H(M). She can then encrypt the digest using a public key encryption function with her secret key and call that the signature. She'll send the message combined with the signature. Then bob can hash the message and decrypt the signature using Alice's public key and be sure that it's from her if they're equal. Since there is no AITM, no one has the facilities to really impersonate either of them.
4.

a. Alice can claim that there was a clash in the hash function used that led to the contract being faulty. I would not believe this since it is known for its reliability and the likelihood of that is super low.(Birthday attack)

b. This would be really silly, but maybe Bob used the wrong cryptographic hash function and this whole thing was just a big misunderstanding. Idk how well I’d believe that considering I’d hold him to higher standards than that, but maybe he truly just messed up.

c. She could claim someone hit him with a replay attack and that his computer automatically rewrote/replaced the original with the modified version, leading him to sue. I could see that happening, especially if it was combined with an extension attack that added executables for replacing the file.

5. Sig_CA = E(S_CA, H("bob.com" || P_B”))
6. She can ask for a digital signature from bob using the same secret key and his public key function. He can follow the typical digital signature protocol of hashing the message and encrypting the digest with his secret key. If she can decrypt it using the same hash function and the public key mentioned in the cert then she’s golden. Maybe?
7. An attacker could

   a. intercept the signature using the public key they could get by asking to talk with bob.com or looking where it’s publicly stored. From there they can write something else and modify the digest so that something looks silly.

   b. Malpractice and or mishandling by the certificate authority could also ruin things, maybe they use an obsolete hashing or during the interaction they have to prove the keys, the attacker does the downgrade thing so they have a guaranteed way in.